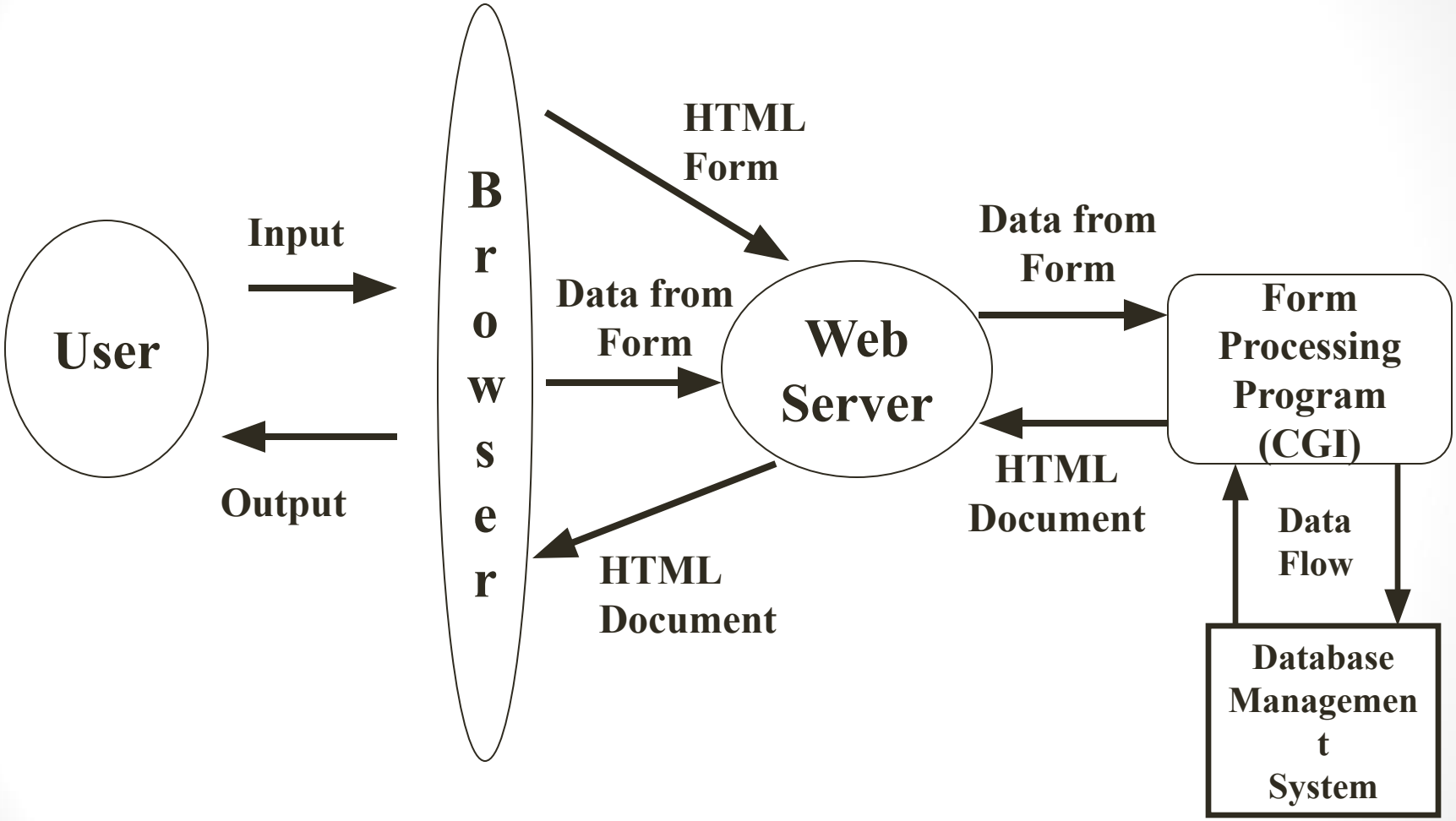
USER INTERACTIONS: FORMS

Form Processing



Flow of Information for Forms

Parts of a Web Form

- A Form is an area that can contain Form Control/Elements.
- Each piece of information for a form is stored in a Field.
- The value itself is called Field Value.

Parts of a Web Form

- Users enter or select a field using Form Control/ Elements.
  - Form/Control elements include: buttons, checkboxes, text fields, radio buttons, drop-down menus, etc
  - A form usually contains a Submit button to send the information in the form elements to the server

Control Elements

- Input Boxes – for text and numerical entries
- Option Buttons, also called Radio Buttons – for selecting a single option from a predefined list
- Selection Lists – for long lists of options, usually appearing in a Drop-Down List Box
- Check Boxes – for specifying yes or no
- Text Areas – for extended entries that can include several lines of text

HTML Forms

- The basic construction of a HTML form is this...

  <form> - begin a form

  <input> - ask for information in one of several different ways

  <input> - there can be as many input areas as you wish

  </form> - end a form HTML form

Forms and Server-Based Programs

- Forms are used to collect information.
- The information is then sent back to the server.
- Information is stored and analyzed using a program on the server.
- By giving users access to programs that react to user input, the Web became a more dynamic environment where companies and users could interact.

Forms and Server-Based Programs

- Server-Based programs provide:
  - Online databases containing customer information
  - Online catalogs for ordering and purchasing merchandise
  - Dynamic Web sites with content that is constantly modified and updated
  - Message boards for hosting online discussion forums

Forms and Server-Based Programs

- Because these programs run on Web servers, rather than locally, users might not have permission to create or edit them. Instead, users will receive information about how to interact with the programs on the server.

- Several reason to restrict direct access:
  - When you run a server-based program, you are interacting directly with the server
    - Security risks (computer hackers)
    - Drain on system resources caused by large number of programs running simultaneously

Forms and Server-Based Programs

- Server-Based Programs
  - Common Gateway Interface (CGI) Scripts
    - Most common
  - ASP
  - Cold Fusion
  - C/C++
  - PHP
  - VBScript
- The Web server determines which language your Web form will use.

Forms

- The FORM element is used to create a data input form.
- A region using forms is enclosed within the <FORM> </FORM> tags.
- A document can have several forms, but the forms should not be embedded.
- The FORM element has three attributes:
  - ACTION, METHOD, and ENCTYPE.

Forms

- **METHOD:**
  - **Specifies the way in which the data from the user are encoded.**
  - **The default METHOD is GET, although the POST method is preferred.**
  - **GET: The CGI program receives the encoded form input in the QUERY_STRING variable, which follows the "?" in the URL that calls the script.**
  - **POST: The CGI script or program receives the encoded form input in its standard input stream. The CONTENT_LENGTH must be used.**

Forms

- **ACTION:**
  - **Specifies the destination URL to which the form should be submitted, once it has been completed by the user.**
  - **If no URL is specified, the URL of the current document containing the form is used.**
  - **MAILTO Action: The data from the form is mailed to the specified E-mail address. Use the POST method.**

Forms

- **ENCTYPE:**
  - **Tell the browser how the data from a form should be encoded when it is returned to the server.**
  - **The default is “application/x-www-form-urlencoded” that converts spaces to “+” and uses “&” to delineated different data fields.**

Getting Started

- The first step in creating a form is to specify the name and location of the CGI script that will be used to process the form data. To do this, type the following code within your HTML file, and note that there are no spaces:
- <form METHOD="Post" ACTION=http://www.temple.edu/cgi-in/mail?*your-e-mail-address@temple.edu*>
- For example, if your e-mail address is jsmith@temple.edu, you would enter:
- <form METHOD="Post" ACTION="http://www.temple.edu/cgi-bin/mail?jsmith@temple.edu ">

Text Input (type="text")

- **A Text Field:**
  - used to create one line fields that viewers can type text. The **default width is 20 characters**, and you can create fields of other sizes by the value in the size option. You can limit the number of characters by the value of the **MAXLENGTH option**. Text input fields will be empty when the page loads, unless you provide an initial text string for its VALUE option

  - **<input type="text" name="textfield" size="value" value="with an initial value">**

Text Input (type="text")

- **Example 1: A text field named "text1" that is 30 characters wide.**
- **<input type="text" name="text1" size="30">**

- **Example 2: A text field named "text2" that is 30 characters wide but will only accept 20 characters.**
- **<input type="text" name="text2" size="30" maxlength="20">**

- **Example 3: A text field named "text3" that is 40 characters wide with default value.**
- **<input type="text" name="text3" size="40" value="We are not alone">**

Password Input (type="password")

- are exactly the same as text input elements, except that when the viewer types in, they see "bullet" characters rather then the letter they are typing. Password text is scrambled during transmission and then unscramble when the form data is received at the server end.

- **Example 4: A password field named "pass1" that is 30 characters wide**
- **<input type="password" name="pass1" size="30">**

- **Example 5: A password field named "pass2" that is 30 characters wide but will only accept 20 characters**
- **<input type="password" name="pass2" size="30" maxlength="20">**

Text Input (type="textarea")

- Text fields that have more than one line and can scroll as the viewer enters more text. The tag options define the size of the field by the number of rows and character columns. By adding the option **WRAP=VIRTUAL**, the text entered will automatically wrap at the right hand side of the field. You can also include default text to appear in the field

- **Example 6: A textarea field named "comments" that is 45 characters wide and 6 lines high**
- **<textarea name="comments" rows="6" cols="45" wrap="virtual">**
  The first time I ever saw a web page, I thought.... (continue)
  </textarea>

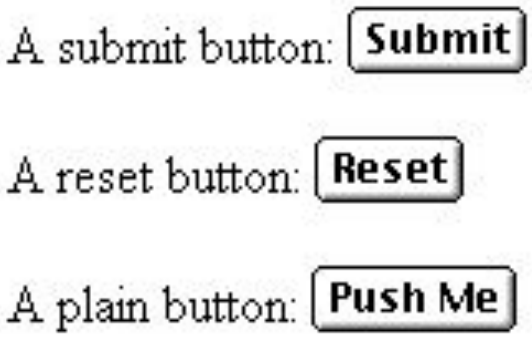
Adding Control Buttons

- A form must include at least one control button for submitting the information once it is filled out. In addition, forms often include a button for resetting all the entries if a person wants to start over.

- When a person presses the submit button, he or she will receive confirmation that the form results were sent to your e-mail address. You will then see an e-mail message in your Inbox with the subject *FORM results*.

Adding Control Buttons

- A submit button:
  <input type="submit" name="Submit" value="Submit">
- A reset button:
  <input type="reset" name="Submit2" value="Reset">



- submit: send data
- reset: restore all form elements to their initial state

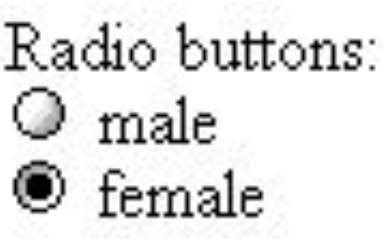
- Note that the type is input, not "button"

Radio buttons (type="radio")

- Are sets of controls that are linked so that only one radio button among each set is selected at a time
- If you click one radio button, the others in the set are automatically de-selected
- A set of radio buttons is defined by providing them the same name
- The value sent in the web form is the value of the radio button that was last selected
- Adding the option CHECKED to one of the buttons in a set will make that button highlighted when the page loads
- Radio buttons do not contain any text

Radio buttons (type="radio")

```
Radio buttons:<br>
  <input type="radio" name="radiobutton" value="myValue1">
  male<br>
  <input type="radio" name="radiobutton" value="myValue2" checked>
  female
```



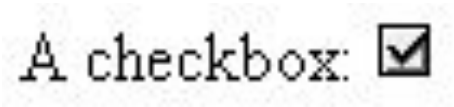
Checkboxes (type="checkbox")

- Are similar to radio buttons, but are not affected by other buttons, so you can have more than one in a group checked at a time
- Note that every checkbox has a unique name. If there is no check in the box, clicking it will place an X or a check mark there
- If the box is checked, clicking it again will remove the mark. The value sent in the web form is the value of the checkbox if it was selected; otherwise the value will be empty
- Adding the option **CHECKED** to a checkbox will make that checkbox highlighted when the page loads.

Checkboxes (type="checkbox")

- A checkbox:

```
<input type="checkbox" name="checkbox"
       value="checkbox" checked>
```

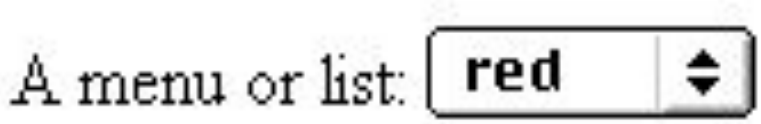


- type: "checkbox"
- name: used to reference this form element from JavaScript
- value: value to be returned when element is checked
- Note that there is *no text* associated with the checkbox—you have to supply text in the surrounding HTML

Drop-down menu or list

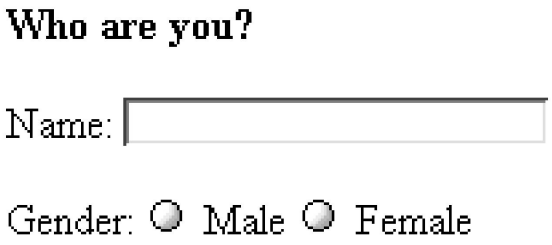
- A menu or list:

```
<select name="select">
    <option value="red">red</option>
    <option value="green">green</option>
    <option value=“blue">blue</option>
</select>
```



  - Additional arguments:
    - size: the number of items visible in the list (default is "1")
    - multiple: if set to "true", any number of items may be selected (default is "false")

Practice Exercise



A complete example

```
<html>
<head>
<title>Get Identity</title>
<meta http-equiv="Content-Type" content="text/html;
        charset=iso-8859-1">
</head>
<body>
<p><b>Who are you?</b></p>
<form method="post" action="">
  <p>Name:
    <input type="text" name="textfield">
  </p>
  <p>Gender:
    <input type="radio" name="gender" value= m >male
    <input type="radio" name="gender" value="f">Female</p>
  </form>
</body>
</html>
```